DIaaS: Data-Intensive workflows as a service - Enabling easy composition and deployment of data-intensive workflows on Virtual Research Environments

ABSTRACT SUBMISSION CONTROL PANEL

Abstract ID#: 156232
Date Started: 2016-08-02 06:17:37
Last Modified: now()
Primary Section/Focus Group: IN

7 Steps to submit an Abstract:

- ✓ 1. Session
- ✓ 2. Abstract Details
- ✓ 3. Author(s)
- ✓ 4. Abstract Text
- ✓ 5. Special Programs
- ✓ 6. Payment
- ✓ 7. Submit Abstract

Options:

Abstract Tutorial

Sign out

-View Submission
-Withdraw
-Transfer
-Letter of Invitation
-Abstracts Viewer
-User Portal

Help:

Report a Technical Issue

Your abstract submission has been received

Click HERE to print this page now.

You have submitted the following abstract to 2016 AGU Fall Meeting. Receipt of this notice does not guarantee that your submission was complete or free of errors.

DIaaS: Data-Intensive workflows as a service - Enabling easy composition and deployment of data-intensive workflows on Virtual Research Environments

Rosa Filgueira[1], Rafael Ferreira da Silva[2], Ewa Deelman[2] and Malcolm Atkinson[1], (1)University of Edinburgh, School of Informatics, Edinburgh, United Kingdom, (2)USC Information Sciences Inst., Marina Del Rey, CA, United States

Abstract Text:

We present the *Data-Intensive workflows as a Service (DIaaS)* model for enabling easy data-intensive workflow composition and deployment on clouds using containers. *DIaaS* model backbone is *Asterism*, an integrated solution for running data-intensive stream-based applications on heterogeneous systems, which combines the benefits of *dispel4py* with *Pegasus* workflow systems. The stream-based executions of an *Asterism* workflow are managed by *dispel4py*, while the data movement between different e-Infrastructures, and the coordination of the application execution are automatically managed by *Pegasus*.

DIaaS combines *Asterism* framework with *Docker* containers to provide an integrated, complete, easy-to-use, portable approach to run data-intensive workflows on distributed platforms. Three containers integrate the *DIaaS* model: a Pegasus node, and an MPI and an Apache Storm clusters. Container images are described as *Dockerfiles* (available online at http://github.com/dispel4py/pegasus_dispel4py), linked to *Docker Hub* for providing continuous integration (automated image builds), and image storing and sharing. In this model, all required software (workflow systems and execution engines) for running scientific applications are packed into the containers, which significantly reduces the effort (and possible human errors) required by scientists or VRE administrators to build such systems. The most common use of *DIaaS* will be to act as a backend of VREs or Scientific Gateways to run data-intensive applications, deploying cloud resources upon request.

We have demonstrated the feasibility of *DIaaS* using the data-intensive *seismic ambient noise cross-correlation* application (Figure 1). The application preprocesses (*Phase1*) and cross-correlates (*Phase2*) traces from several seismic stations. The application is submitted via Pegasus (*Container1*), and *Phase1* and *Phase2* are executed in the MPI (*Container2*) and Storm (*Container3*) clusters respectively. Although both phases could be executed within the same environment, this setup demonstrates the flexibility of *DIaaS* to run applications across e-Infrastructures.